



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DESARROLLO DE UN MAP BLOG USANDO EL API DE GOOGLE MAPS
PARA GEOREFERENCIAR LUGARES O EVENTOS QUE HAYAN
OCURRIDO EN EL ECUADOR”**

TESINA DE SEMINARIO

Previa a la obtención del Título de:

**INGENIERO EN COMPUTACIÓN
ESPECIALIZACIÓN
SISTEMAS TECNOLÓGICOS**

Presentado por:

GONZALO GABRIEL GARCÍA GAVILANES

Guayaquil - Ecuador

2010

AGRADECIMIENTO

*A Dios,
por haberme dado la fortaleza necesaria para superar los obstáculos.*

*A mi esposa,
Por su inmenso amor y cariño, y esa valiosa confianza depositada en mí*

*A mis padres,
por brindarme un hogar cálido y los consejos entregados para ser una
persona de bien.*

*A mis hermanos,
por ser muy buenos amigos y estar cuando los he necesitado*

*Al Ingeniero Fabricio Echeverría,
por sus valiosas enseñanzas que me han permitido seguir creciendo
intelectualmente.*

DEDICATORIA

*Con mucho cariño,
a mi familia y amigos,
por su apoyo incondicional.*

TRIBUNAL DE SUSTENTACIÓN

Ing. Jorge Aragundi R.
Presidente

Ing. Fabricio Echeverría B.
Profesor de Seminario

Ing. Cristina Abad R.
Profesor Delegado del Decano

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo Final de Graduación, me corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de Pregrado de la ESPOL).

Gonzalo Gabriel García Gavilanes

RESUMEN

En el presente trabajo de graduación se detalla el desarrollo de un aplicativo Web llamado Map Blog, el mismo fue desarrollado con el API de Google Maps y permite a los usuarios ubicar en un mapa algún lugar, noticias y/o evento que ocurra en el Ecuador agregar una entrada (post) y subirlo al sistema para que sea visto y comentado por otros usuarios.

El capítulo 1 presenta una breve introducción al marco teórico del proyecto. Se ve también cuál es el alcance y objetivo de fusionar estas dos aplicaciones para formar el MAP BLOG y presentamos las implementaciones existentes de Web Blogs y Google Maps.

En el capítulo 2 se conoce todo acerca del análisis y diseño del Map Blog. En primera instancia se presenta los pasos que necesitan para poder agregar un mapa de Google en nuestro sitio Web, luego se muestra cómo usar el API y sus funciones básicas, cómo trabaja el Map Blog para obtener las imágenes de los mapa, se da una explicación del funcionamiento de cada interfaz y finalmente se observa una definición de las tablas relevantes del proyecto.

En el capítulo 3 encontramos información sobre las herramientas utilizadas para la implementación, y las consideraciones que se tuvieron para realizar las pruebas.

Las conclusiones y recomendaciones sobre el sistema se mencionan al final de este documento.

ÍNDICE GENERAL

AGRADECIMIENTO.....	ii
DEDICATORIA.....	iii
TRIBUNAL DE SUSTENTACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN.....	vi
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	x
INTRODUCCIÓN.....	1
1. API DE GOOGLE MAPS	2
1.1. Definición	2
1.1.1. Google Maps	2
1.1.2. Blogs en la Web.....	3
1.2. Motivación	3
1.3. Características.....	4
1.3.1. Google Maps	4
1.3.2. Blogs o Weblogs	5
1.4. Objetivo	5
1.5. Implementaciones existentes de Google Maps y Blogs	5
2. ANÁLISIS Y DISEÑO DEL MAP BLOG	7
2.1. ¿Cómo trabajar con Google Maps?	7
2.1.1. Obtener la Key de Google Maps	7
2.1.2. Creando el contenedor del mapa	9

2.1.3.	¿Cómo usar el API?	9
2.2.	<i>Análisis</i>	12
2.3.	<i>Diseño del Map Blog</i>	13
2.3.1.	Funcionamiento	14
2.4.	<i>Modelo de Datos</i>	15
2.5.	<i>Diseño de interfaz con el usuario</i>	16
2.5.1.	Tipos de usuario	16
2.5.2.	Flujo de ventanas	17
2.5.3.	Diseño de la interacción del usuario	18
3.	IMPLEMENTACIÓN Y PRUEBAS	27
3.1.	<i>Herramientas de desarrollo utilizadas</i>	27
3.2.	<i>Pruebas</i>	29

CONCLUSIONES Y RECOMENDACIONES

REFERENCIAS BIBLIOGRÁFICAS

GLOSARIO

ÍNDICE DE FIGURAS

Figura 1-1 Mapa de Google	4
Figura 1-2 Blog de la Espol desarrollado con Wordpress	5
Figura 1-3 Anieto2K desarrollado con Wordpress.....	6
Figura 1-4 Sitio de Google Maps	6
Figura 1-5 Sitio de Desarrollo Web.....	6
Figura 2-1 Formulario para obtener clave de Google Map.....	8
Figura 2-2 Código de javascript que da inicialmente Google para cargar los mapas	11
Figura 2-3 Funcionamiento Map Blog.....	14
Figura 2-4 Modelo de Datos.....	15
Figura 2-5 Flujo Suscriptor.....	17
Figura 2-6 Flujo Autor	18
Figura 2-7 Flujo Administrador.....	18
Figura 2-8 Pagina inicial del Map Blog	19
Figura 2-9 Formulario de Registro	20
Figura 2-10 Formulario de inicio de sesión.....	21
Figura 2-11 Pantalla de administrador para publicar las entradas	22
Figura 2-12 Entradas Recientes	23
Figura 2-13 Entradas Recientes - Vista del Mapa	23
Figura 2-14 Ver Comentarios.....	24
Figura 2-15 Búsqueda de Lugares	25
Figura 2-16 Cambiar Mi Avatar	26

INTRODUCCIÓN

En la actualidad los Blogs o Weblogs usualmente nos restringen solamente a subir comentarios acerca de un artículo o noticia; así mismo son escasos los aplicativos Web que implementen Google Maps en sus sitios Web, y de hacerlo los mapas suelen presentar información estática.

Este trabajo de graduación pretende conjugar ambos conceptos en un solo aplicativo y cambiar el paradigma de los blogs, permitiendo a los usuarios buscar y ubicar en un mapa de Google los lugares, eventos y/o noticias suscitados en el Ecuador, para luego crear una nueva entrada y que sea comentada por demás usuarios, haciendo que de esta forma la información reflejada en los mapas sea dinámica una vez que sea consultada por los usuarios registrados en el sistema.

El objetivo es crear un aplicativo web, que permita a los usuarios manipular la información creada, realizar consultas de eventos y hechos ocurridos, crear sus propias entradas en un blog permitiéndole a más de estos ubicarlas en un mapa de Google.

CAPÍTULO I

1. API DE GOOGLE MAPS

1.1. Definición

1.1.1. Google Maps

Google dentro de sus amplios servicios gratuitos tiene a **Google Maps** como un API el cual nos permite insertar mapas en nuestras aplicaciones Web, brindándonos diversas utilidades y eventos para manipularlos. Dichas imágenes son presentadas como fotos satelitales del mundo entero. Podemos encontrar un listado de los servicios ofrecidos por Google en (Spacebom).

Fue anunciado el 8 de febrero de 2005 y la (Wikipedia) nos indica que el API de Google Maps desde sus inicios soportaría solo a los usuario de Internet Explorer y Firefox, pero el soporte para los demás Web browsers fue agregado el 25 de febrero de 2005. Dicho API funciona a base de Javascript, por lo cual puede ser implementado en páginas estáticas usando HTML o en páginas Web dinámicas ya sean estas JSP, PHP, ASP, entre otras.

1.1.2. Blogs en la Web

La enciclopedia libre (WikipediaBlog) nos dice que un Blog o bitácora es un sitio Web que periódicamente es actualizado, el cual recopila de manera cronológica textos o artículos de uno o varios autores, apareciendo primero el más reciente.

El sitio (Minid.net) nos indica que una característica muy clara que tiene el Weblog, que los diferencia de los demás sitios Web es su cronología, ya que da un aspecto de diario personal, dotando al Weblog de un orden claro y lógico lo cual es importante para buscar los temas tratados de una manera mucho más rápida. Los Weblogs están estructurados de la siguiente forma:

- Fecha
- Título
- Cuerpo de la entrada
- Categoría
- Comentarios

1.2. Motivación

Actualmente la mayoría de los blogs o Weblogs encontrados en la Internet, nos permiten crear una entrada sobre algún artículo o noticia, y hacer comentarios sobre ellos. Al realizar este proyecto buscamos crear un aplicativo Web 2.0 que conjugue los conceptos de un Weblog

y los mapas de Google para lo cual se va usar el API creado para tal efecto, con lo que surge el Map Blog Espol.

1.3. Características

1.3.1. Google Maps

Google Maps brinda la capacidad de poder hacer acercamientos a la superficie del mapa, cambiar los modos de vista básicos: Mapa, Satélite e Híbrido, permite desplazamiento sobre el mapa usando el mouse, búsqueda de direcciones y ubicarlas en el mapa. A más de estas características básicas se pueden programar diferentes eventos sobre el mapa, tales como el click, move, etc.



Figura 1-1 Mapa de Google

1.3.2. Blogs o Weblogs

Un Blog o Weblog es un formulario que permite a otros usuarios escribir sus comentarios, agregar enlaces externos, imágenes y puede ser realizado sobre un editor de texto enriquecido.

1.4. Objetivo

Desarrollar un aplicativo Web, que permita georeferenciar los diferentes eventos que hayan ocurrido en el Ecuador y/o ubicar algún lugar determinado en el Mapa, y a su vez permita subir comentarios referentes a dichos eventos o establecimientos.

1.5. Implementaciones existentes de Google Maps y Blogs

Hoy en día existen muchas (Herramientas) que prestan todas las bondades para implementar un blog de una manera rápida y eficiente, entre ellas está (Wordpress) un software que permite administrar los blogs creados directamente en el servidor que aloja el sitio, entre los sitios que están implementados con Wordpress tenemos:



Figura 1-2 Blog de la Espol desarrollado con Wordpress



Figura 1-3 Anieto2K desarrollado con Wordpress

En lo que se refiere a implementaciones que usen el API de Google Maps la única referencia encontrada es la propia de Google. Ya que la mayoría de sitios encontrados de Google Maps nos indican solamente cómo usar el API.



Figura 1-4 Sitio de Google Maps



Figura 1-5 Sitio de Desarrollo Web

CAPÍTULO II

2. ANÁLISIS Y DISEÑO DEL MAP BLOG

2.1. ¿Cómo trabajar con Google Maps?

Como ya se indicó anteriormente para trabajar con el API se necesita conocer de javascript y se debe realizar los siguientes pasos básicos tal como cita (Google)

- Se obtiene una Key de Google Maps.
- Creamos el contenedor del mapa.
- Se escribe una función JavaScript para crear un objeto de mapa e incluimos el código del API de Google Maps mediante la etiqueta ***script***.
- El mapa debe ser centrado en un punto geográfico concreto.
- Se inicializa el objeto de mapa desde el evento ***onLoad*** de la etiqueta ***body***.

2.1.1. Obtener la Key de Google Maps

Inserta un mapa en un sitio Web es muy simple. Lo primero que se debe hacer es solicitar un **Key** para poder usar el API, para lo cual previamente debemos disponer de un correo gmail, por ejemplo: micorreo@gmail.com. Se Ingresa a la dirección de

Internet <http://code.Google.com/intl/es-ES/APIs/maps/signup.html>, dicha página nos pedirá la dirección URL de nuestro sitio Web donde serán mostrados los mapas.

A screenshot of the Google Maps API signup form. It features a checkbox labeled 'He leído y acepto los términos y condiciones' with a link to '(versión imprimible)'. Below this is a text input field for 'Dirección URL de mi sitio web' containing the value 'http://127.0.0.1:8989/MapBlogApp/'. At the bottom is a button labeled 'Generar clave de API'.

Figura 2-1 Formulario para obtener clave de Google Map

El API de Google Maps tiene sus restricciones en cuanto a su uso entre las que destacan las principales:

- El API de Google Maps no incluye publicidad.
- No puedes modificar ni ocultar los logotipos ni la atribución en el mapa.
- No se puede indicar en el mapa lugares donde se expendan cualquier tipo de drogas o estupefacientes.

Al momento de generar la **Key**, Google nos dará algo parecido a lo siguiente.

A blue rectangular box containing a long alphanumeric string representing a Google Maps API key: ABQIAAAA6ztTU_R6Fg4G5z_F3u6ARhSPkltHjFzcPpAuFBLD7lzg_V9yVxT5sjlmZFPuFjzGZ0ASXR0mxMDYYg.

Dicha key es única y valedera para el URL ingresado previamente.

2.1.2. Creando el contenedor del mapa

Para que el mapa se muestre en nuestra página Web, se debe reservar un espacio para éste, eso lo realizamos creando un elemento **div** al cual le damos un id para ser accesado mediante DOM.

```
<div id="map_canvas" style="width: 500px; height: 300px">
</div>
```

2.1.3. ¿Cómo usar el API?

Google Maps nos genera un bloque de código en Javascript, con lo que podemos darnos cuenta que solo necesitamos saber javascript para hacer uso de este API.

```
<script
src="http://maps.google.com/maps?file=api&v=2&sensor=true_or_f
alse&key=ABQIAAAA6ztTU_R6Fg4G5z_F3u6ARhSPkltHjFzcPpAu
FBLD7Izg_V9yVxT5sjlmZFPuFjzGZ0ASXR0mxMDYYg"
type="text/javascript">
</script>
```

Vamos a analizar el código de arriba y daré una breve descripción de que significa cada parámetro que se pasa en el query string del atributo **src**.

- file: Indica que vamos a usar los métodos y clases del API de Google Maps.
- v: Nos indica la versión del API a usarse, actualmente ya está la versión 2.75, si la queremos usar basta con cambiar el parámetro v=2 por v=2.x con lo que tomará la última versión del API, aunque más estable es la versión 2 v=2.
- sensor: Nos indica si nuestro aplicativo va a recibir coordenadas mediante algún dispositivo como un GPS, de ser así nuestro aplicativo debería tener sensor=true y false en caso contrario.
- key: Aquí colocamos el **Key** proporcionado por Google.

Posteriormente se procede a escribir el código javascript el cual instanciará la clase **GMap2**, la cual posee todo lo necesario para inicializar el mapa. Como parámetro recibe el contenedor HTML donde se mostrará nuestro Google Map. Una vez realizado esto, debemos hacer que nuestro mapa se presente en un punto determinado indicándole la latitud y longitud inicial.

```
<script type="text/javascript">
function initialize() {
  if (GBrowserIsCompatible()) {
    var map =
new GMap2(document.getElementById("map_canvas"));
    map.setCenter(new GLatLng(37.4419, -122.1419), 13);
  }
}
</script>
```

Figura 2-2 Código de javascript que da inicialmente Google para cargar los mapas

Este código debe ir en la entre los tags **<head></head>** de nuestra página HTML. Ahora analizaremos el código de ejemplo brindado por Google para cargar nuestros mapas en nuestros contenedores HTML.

GBrowserIsCompatible, nos devuelve verdadero si el navegador utilizado es compatible con Google Maps. A continuación se muestra una lista de los navegadores que son compatibles con el API.

- IE 6.0+
- Firefox 0.8+
- Safari 1.2.4+
- Netscape 7.1+
- Mozilla 1.4+
- Opera 8.02+

Una vez que se ha comprobado la compatibilidad del navegador creamos el objeto **GMap2**, pasándole como parámetro el id de nuestro contenedor div en el cual se va a cargar el mapa de Google.

Y por ultimo centramos el mapa usando el método **setCenter**, pasando como parámetro una latitud y longitud por medio de la clase **GLatLng**.

Cuando el script este agregado en nuestra página Web invocamos nuestra función **initialize()** desde el elemento body.

```
<body onload="initialize()" onunload="GUnload()">
```

2.2. Análisis

En base a lo expuesto en el primer capítulo de este documento (Implementaciones existentes de Google Maps y Blogs), de que solo existen Blog o Web Blogs para postear y/o comentar artículos o noticias y no existen aplicaciones de Google Maps que los usuarios puedan manipular, se ha pensado en desarrollar el Map Blog, que contemple estos dos conceptos en un solo aplicativo Web que use el API de Google Maps para poder ubicar las noticias en los mapas dados por Google.

El Map Blog es un aplicativo Web 2.0, que permite llevar control de la creación de todas las entradas y comentarios, para lo cual se registra en una base de datos quién subió dicha información.

Existe una opción para los usuarios administradores, en la cual se analizan las entradas subidas antes de su publicación, ya que se debe revisar detenidamente la información publicada.

2.3. Diseño del Map Blog

El Map Blog fue diseñado de tal forma que pueda ser soportado por cualquier Application Server existente en el mercado, así mismo pueda acoplarse a cualquier base de datos transaccional, y para cargar los mapas de Google es necesario disponer de Internet. A continuación se muestra una representación de cómo funciona el Map Blog.

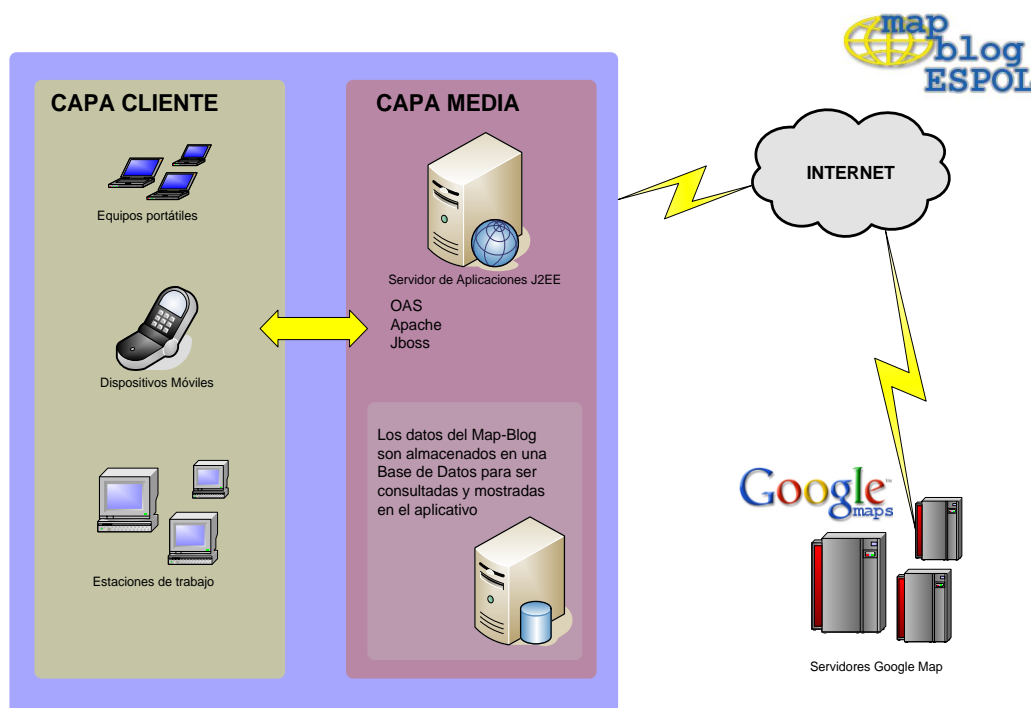


Figura 2-3 Funcionamiento Map Blog

2.3.1. Funcionamiento

La capa cliente se conecta al sitio al Map Blog almacenado en un Web Server, este a su vez como ya dispone de un Key válido para la URL que va a contener los mapas, se conecta mediante Internet a los servidores de mapas de Google y obtiene como respuesta las imágenes de los mapas, con las coordenadas que éste es inicializado.

Posteriormente, el cliente tiene la facilidad de moverse en el mapa y ubicar el lugar visitado o el evento ocurrido (internamente el aplicativo obtiene las coordenadas del punto), luego el post es almacenado y está a la espera que el Administrador lo apruebe y sea publicado y pueda ser comentados por los demás usuarios.

A partir de esta funcionalidad el cliente podrá hacer uso del Map Blog para lo cual debe disponer de un usuario y una contraseña.

2.4. Modelo de Datos

El motor de base de datos que se usó como repositorio de datos es MySql 5.0.

A continuación, se da a conocer el modelo de datos del Map Blog, del cual se explicará las tablas relevantes para el proyecto y necesarias para almacenar los usuarios, datos de las entradas, los comentarios relacionados a cada una y las coordenadas del mapa.

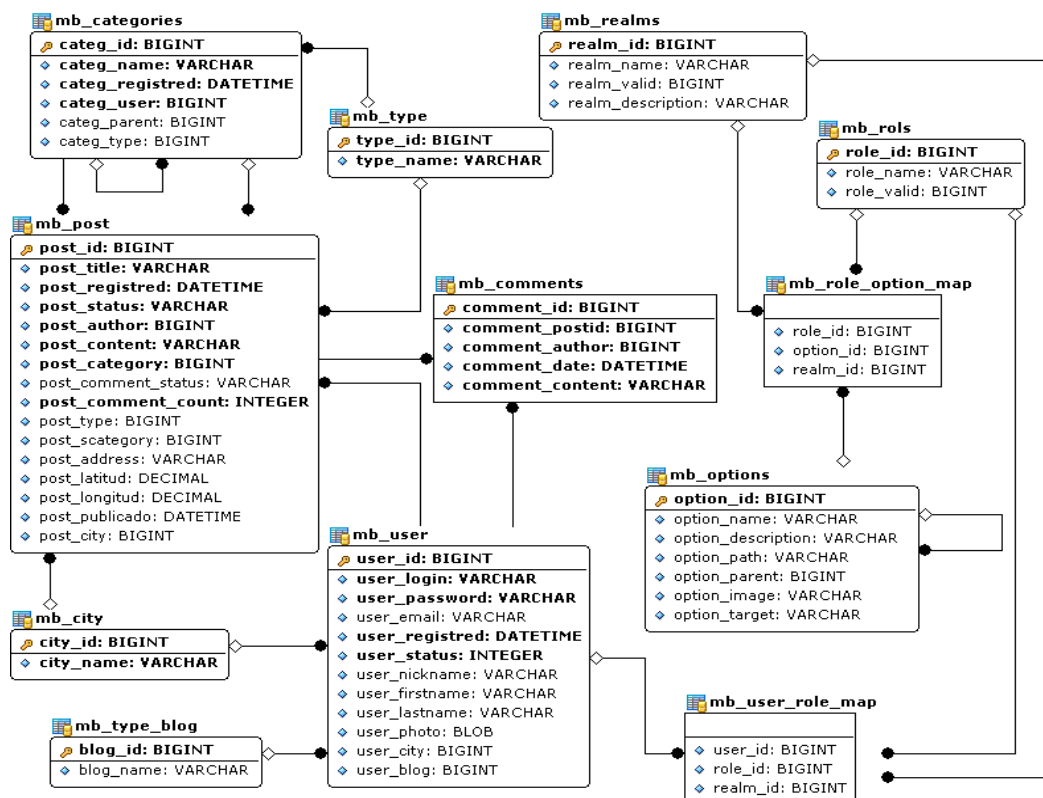


Figura 2-4 Modelo de Datos

- **mb_post:** Permite almacenar el los datos de la entrada desde que se crea. Entre sus datos están los campos de latitud y longitud que almacenan las coordenadas del punto seleccionado por el usuario.
- **mb_comments:** Permite almacenar los comentarios que los usuarios han creado después de que se publicó la entrada.
- **mb_categories:** Almacena las categorías y subcategorías de las entradas creadas; identifican de qué clase son, para una mejor búsqueda
- **mb_user:** Almacena los usuarios que se registran en el Map Blog. A cada usuario registrado por defecto se le asigna el rol de suscriptor.

2.5. Diseño de interfaz con el usuario

2.5.1. Tipos de usuario

El Map Blog contempla 3 tipos de usuario, entre los cuales tenemos:

- **Suscriptor:** Es el tipo de usuario con el que se registra el usuario por defecto. Solo puede buscar lugares ya publicados, consultar las entradas recientes y subir comentarios sobre los mismos.

- Autor: Igual que el suscriptor pero este tiene la opción de poder crear las entradas, las cuales no serán publicadas hasta que sean analizados por el administrador del sistema.
- Administrador: Tiene los privilegios del autor, y además puede analizar las entradas creadas por el autor y permitir su publicación.

2.5.2. Flujo de ventanas

El flujo de ventanas describe la secuencia de las páginas que el usuario observará según el tipo que éste posea.

En las siguientes figuras veremos el flujo de ventanas para cada tipo de usuario.



Figura 2-5 Flujo Suscriptor

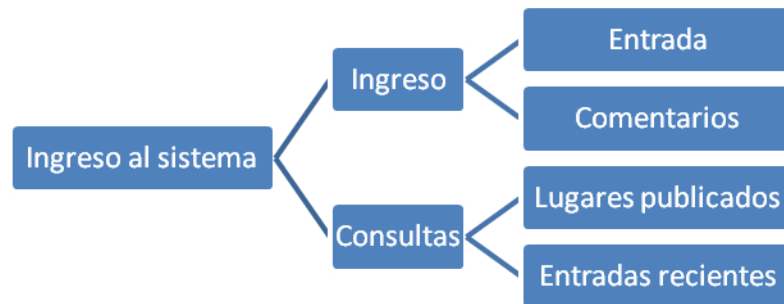


Figura 2-6 Flujo Autor



Figura 2-7 Flujo Administrador

2.5.3. Diseño de la interacción del usuario

En este punto se presenta el diseño de las páginas del Map Blog desarrolladas para la demostración de este proyecto.

En primer lugar veremos que para todos los usuarios que no han iniciado sesión se les muestra un menú superior, y un menú de usuario público del lado izquierdo. Una vez que se inicia la sesión, se carga un menú dependiendo del tipo de usuario.

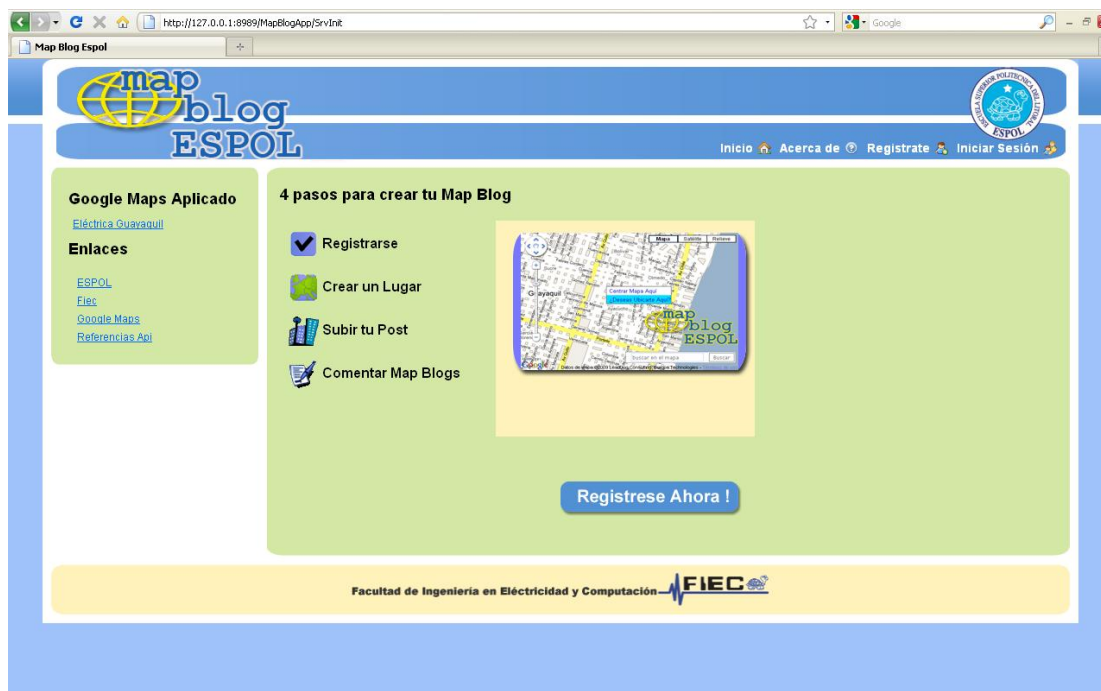


Figura 2-8 Pagina inicial del Map Blog

Toda las demás información, tal como formularios y datos del sistema son mostrados en la parte central de la página.

El formulario de registro permite a los usuarios nuevos registrar sus datos para poder ingresar al Map Blog. Entre los datos que se solicitan tenemos:

Nombres, apellidos, email, usuario, clave, confirmación de clave, Nick para mostrar en las entradas creadas o comentarios realizados, la ciudad de residencia y el avatar que es una imagen que identifica al usuario, este dato puede ser obviado.

The image shows a web browser window displaying the registration page for 'Map Blog ESPOL'. The browser's address bar shows the URL 'http://127.0.0.1:8989/MapBlogApp/usuario/registrarse.jsp'. The page has a blue header with the 'map blog ESPOL' logo on the left and navigation links 'Inicio', 'Acerca de', 'Registrate', and 'Iniciar Sesión' on the right. A circular ESPOL logo is also present in the top right corner. The main content area has a light green background and is divided into two sections. On the left, under the heading 'Google Maps Aplicado', there is a link 'Eléctrica Guayaquil' and a section titled 'Enlaces' with links to 'ESPOL', 'Fiec', 'Google Maps', and 'Referencias Api'. On the right, there is a registration form with a large orange 'Registrarse' button at the top. Below this button are input fields for 'Nombres', 'Apellidos', 'Email', 'Usuario', 'Clave', and 'Confirmar Clave'. There is also a 'Nick a Mostrar' field, a 'Ciudad' dropdown menu with the text 'Seleccione su Ciudad', and a 'Mi Avatar' field. A small 'Examinar...' button is next to the 'Mi Avatar' field. At the bottom of the form area is another 'Registrarse' button. The footer of the page is yellow and contains the text 'Facultad de Ingeniería en Electricidad y Computación' followed by the 'FIEC' logo.

Figura 2-9 Formulario de Registro

El formulario de iniciar sesión, nos permite ingresar al Map Blog una vez que el usuario se haya registrado en el sistema.

Dicho usuario se le asignará automáticamente el rol de suscriptor, para lo cual se le cargará el menú correspondiente, tal y como se muestra en la figura siguiente.

The image shows a web browser window displaying the login page of 'Map Blog ESPOL'. The browser's address bar shows the URL 'http://127.0.0.1:8989/MapBlogApp/usuario/login.jsp'. The page has a blue header with the 'map blog ESPOL' logo and navigation links: 'Inicio', 'Acerca de', 'Regístrate', and 'Iniciar Sesión'. On the left, there is a sidebar with 'Google Maps Aplicado' and 'Enlaces' (ESPOL, Fiec, Google Maps, Referencias Aoi). The main content area has an orange 'Iniciar Sesión' button at the top, followed by input fields for 'Usuario' and 'Clave', and an 'Ingresar' button. The footer includes the text 'Facultad de Ingeniería en Electricidad y Computación' and the 'FIEC' logo.

Figura 2-10 Formulario de inicio de sesión

La opción de Mantenimiento (Entradas por Aprobar), es únicamente asignada al administrador, ya que este usuario es el único que puede publicar las entradas subidas por los autores. El post podrá ser publicado siempre y cuando este haya sido analizado.

Para publicar cada post se debe dar click en el link **Permitir Publicación**, con lo que se cambia el estado del post a publicado y podrá ser visto por los demás usuarios.



Figura 2-11 Pantalla de administrador para publicar las entradas

La opción Recientes (Entradas Recientes), puede ser usada por cualquier tipo de usuario; permite ver todas las entradas que han sido publicadas para poder realizar cualquier tipo de comentario sobre ésta.

Para subir los comentarios se dispone de un editor de texto enriquecido, el cual nos permite crear texto con formato. A más de esto nos da la opción de tener una vista del mapa (Ver Mapa).

Figura 2-13 Entradas Recientes - Vista del Mapa

En la opción Recientes se encuentra el link (Ver comentarios el cual nos muestra, de manera cronológica todos los comentarios que han sido creados para una entrada.

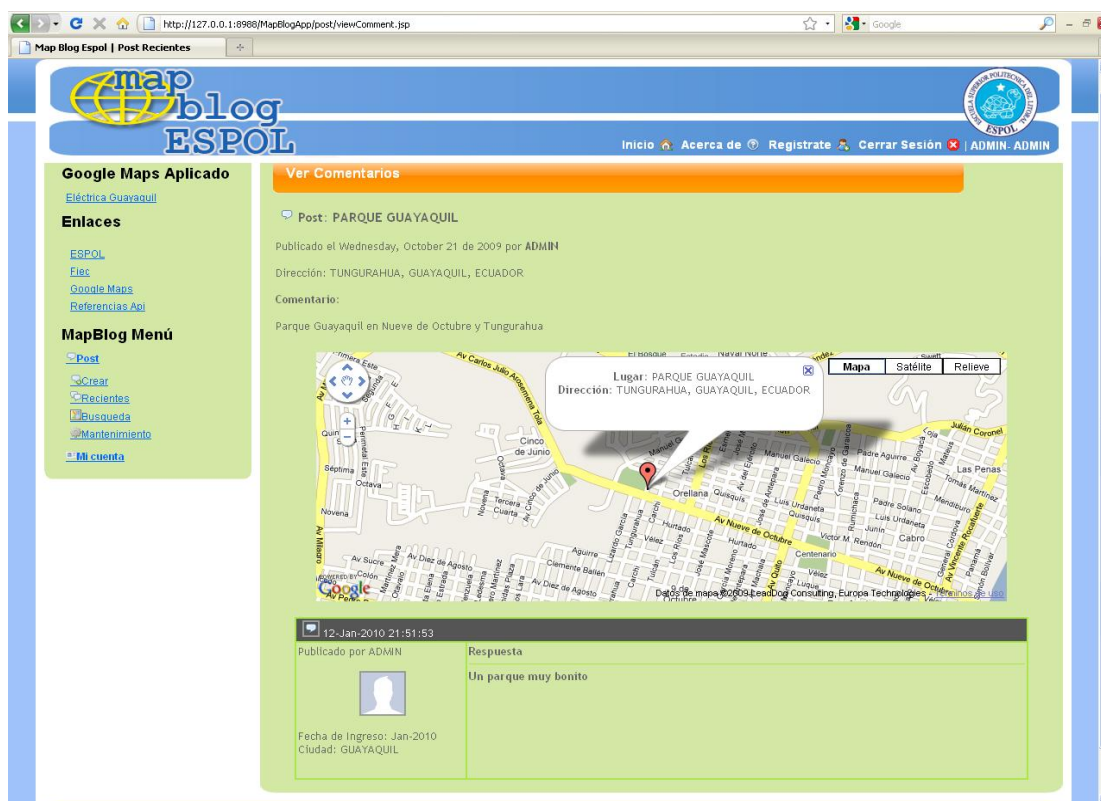


Figura 2-14 Ver Comentarios

La opción Búsqueda (Búsqueda de Lugares), puede ser usada por cualquier tipo de usuario; permite mediante un árbol ver todas las categorías de las que dispone el Map Blog para agilizar y facilitar la búsqueda de lugares e información a los usuarios del sistema.

Al seleccionar la categoría y subcategoría, inmediatamente se visualiza la información ingresada con estos parámetros para ser ubicados en el mapa.

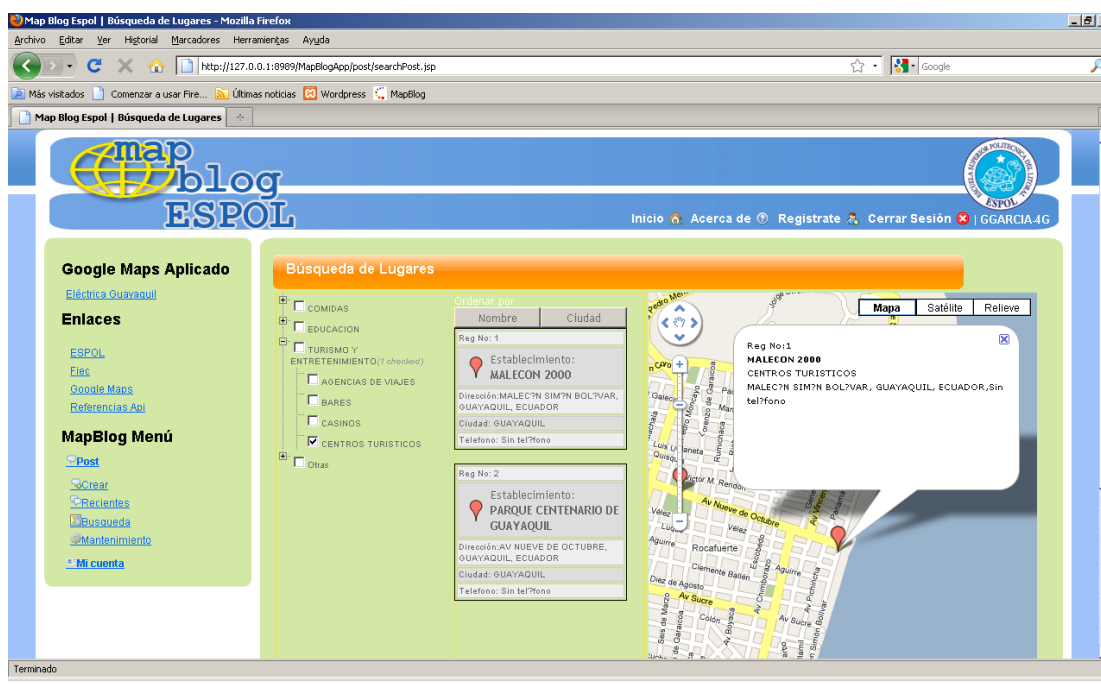


Figura 2-15 Búsqueda de Lugares

En el menú Mi cuenta se encuentra la opción Cambiar Avatar, la cual permite subir tu imagen del blog si no fue ingresada durante el registro de usuario o modificar la existente.

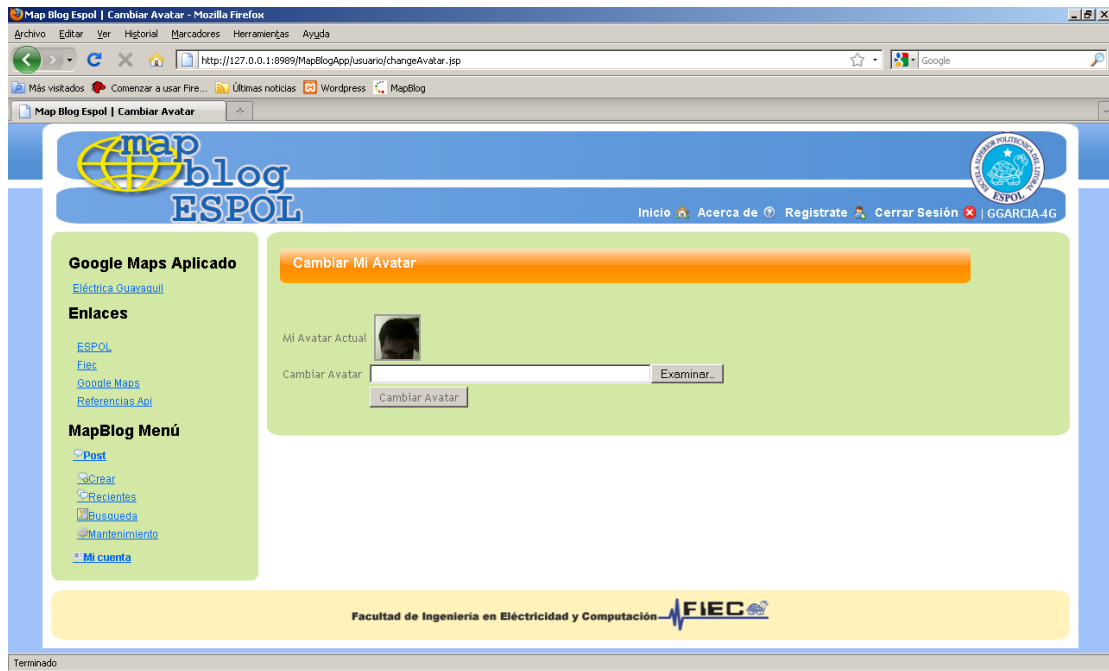


Figura 2-16 Cambiar Mi Avatar

CAPÍTULO III

3. IMPLEMENTACIÓN Y PRUEBAS

3.1. Herramientas de desarrollo utilizadas

A continuación se detallarán las herramientas usadas para el desarrollo del proyecto.

Para la capa cliente o de presentación se eligió trabajar con el lenguaje de scripting JSP. Como IDE para el desarrollo de las páginas y clases de java se usó Oracle JDeveloper 11g. Para poder usar el API de Google Maps se usó Javascript y CSS para los estilos.

Para la capa media se usó Jboss 4.2.3.GA como Application Server, aunque el aplicativo está diseñado para ser soportado por cualquier servidor de aplicaciones J2EE. Para el core se usó lenguaje Java y JDK 1.5. Como repositorio de datos se decidió trabajar con MySql 5.0.

Existe un archivo de configuración XML llamado ConfigMapBlog.xml, que contiene datos tales como la clave de Google Maps, los parámetros de la base de datos a la cual se conecta, de tal manera que es posible conectarse a cualquier repositorio de datos, siempre y cuando se migre la metadata y se posea los parámetros de conexión a

la base de datos. A continuación se presenta la estructura del archivo de XML de configuración.

```
<?xml version = "1.0" encoding = "iso-8859-1"?>
```

```
<config>
```

```
    <googleKey>ABQIAAAA6ztTU_R6Fg4G5z_F3u6ARhRPKwivG  
    KVDfcBoezjz6aJLr6dFKhTSyC3gDiC0OJ-  
    cndFamg_AEKXnzw</googleKey>
```

```
    <appserver>jboss</appserver>
```

```
    <vendor>mysql</vendor>
```

```
    <datasource></datasource>
```

```
    <server>191.9.202.21</server>
```

```
    <port>3306</port>
```

```
    <username>mapblog</username>
```

```
    <password>mbroot</password>
```

```
    <database>MapBlogBD</database>
```

```
    <instance></instance>
```

```
</config>
```

3.2. Pruebas

En esta sección se hace una descripción general de las pruebas realizadas al sistema para garantizar su correcto funcionamiento.

En primer lugar se realizaron las pruebas de unidad, las que se centraron en cada opción implementada en el código para probar que el flujo de cada uno sea el correcto. Para esto se tomaron en cuenta los siguientes aspectos:

- Se debe asegurar que el flujo de información se produzca de manera adecuada a través de la interfaz del sistema.
- Las estructuras de base de datos de cada opción deben conservar los datos que contienen durante toda la ejecución de la opción.
- Los puntos seleccionados en el mapa sean almacenados correctamente, así mismo que al momento de hacer las consultas estos puntos sean pintados en las coordenadas correctas.
- Se probó que sean cargadas las opciones correspondientes dependiendo del tipo de usuario.
- Que la imagen o avatar sea almacenada íntegramente en la base de datos.

- Probar los casos de error que pueden suscitarse durante la ejecución del sistema.
- Controlar las inconsistencias que pueden haber en los tipos de datos.

De esta forma se comprobó que el sistema cumpla con los requerimientos bajo los cuales fue desarrollado, cubriendo de esta manera las pruebas de validación del sistema.

Finalmente se probó con otros usuarios la interacción con el sistema; esto es, que la interfaz desarrollada sea fácil de usar y comprensible en su manejo y funcionalidad. Para esto se usaron nombre apropiados a cada opción, alertas cuando ocurran errores y mensajes de información por cada transacción.

CONCLUSIONES Y RECOMENDACIONES

Al culminar con este proyecto de graduación pude concluir:

1. Que es muy importante el desarrollo de aplicaciones Web con APIs que permitan realizar aplicaciones más robustas que le den la facilidad al usuario de crear su propio contenido.
2. El Google Map es uno de los procesadores de mapas más eficientes, ya que el mercado existen otros como el uMapper que no permiten tener la vista de los diferentes tipos de mapa que se pueden apreciar como la vista Satelital.
3. La Web 2.0 esta ganado terreno en lo que se refiere al desarrollo sitios Web transaccionales, ya que con los APIs existentes se puede obviar ciertos desarrollos extras como por ejemplo un mecanismo de autenticación (login), o búsquedas. De esta forma se pueden desarrollar de manera más rápida nuestros sitios Web.
4. Dichos API pueden ser usados con cualquier tipo de lenguaje de scripting.

Se recomienda que para un futuro:

1. Que el aplicativo sea desarrollado para dispositivos móviles, ya que las funciones usadas no son soportadas en estos dispositivos, esto en cuanto a los javascript (capa cliente).

2. Que el aplicativo pueda ser parametrizado, para que de alguna forma se permita recibir las coordenadas desde un dispositivo GPS.
3. Que en el mapa se pueda delimitar una región mediante Líneas de Polígonos, que es una clase que forma parte del API de Google Maps; esto con el fin de poder marcar una región extensa.

REFERENCIAS BIBLIOGRÁFICAS

[1] **Google**, Concepto básicos de los mapas - Api de Google Maps,
[http://code.google.com/intl/es-](http://code.google.com/intl/es-ES/apis/maps/documentation/introduction.html#Loading_the_Maps_API)

[ES/apis/maps/documentation/introduction.html#Loading_the_Maps_API](http://code.google.com/intl/es-ES/apis/maps/documentation/introduction.html#Loading_the_Maps_API) , 2009.

[2] **Herramientas**, Blog - Wikipedia, enciclopedia libre, <http://es.wikipedia.org/wiki/Blog> ,
2009.

[3] **Minid.net**. Minid.net >> ¿Qué es un Weblog?, <http://www.minid.net/que-es-un-weblog/>
, 2009.

[4] **Spacebom**. Lista de Servicios de Google Inc,
<http://blog.spacebom.com/22/del/11/del/2005-248>, 2009.

[5] **Wikipedia**. Google Maps, Wikipedia, http://es.wikipedia.org/wiki/Google_Maps , 2009.

[6] **WikipediaBlog**. Blog - Wikipedia, la enciclopedia libre, <http://es.wikipedia.org/wiki/Blog> ,
2009.

[7] **Wordpress**. Wordpress, <http://wordpress.org> , 2009.

GLOSARIO

API. Application Programming Interface.

Application Server. Framework dedicado a la ejecución de procedimientos, rutinas y scripts de un aplicativo web.

DOM. Document Object Model.

Editor de Texto Enriquecido. Herramienta de Web Browser que permite sustituir un textarea de HTML por un editor como Word.

GPS. Global Position System.

HTML. HyperText Markup Language.

IDE. Integrated Development Environment.

JDK. Java Development Kit.

JSP. Java Server Pages.

PHP. Hyper Text Pre-Processor.

URL. Uniform Resource Locator.